Troubleshooting Guide

Oracle FLEXCUBE Onboarding

Release 14.5.2.0.0

Part Number F46759-01

August 2021

ORACLE®

Troubleshooting Guide

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax: +91 22 6718 3001

https://www.oracle.com/industries/financial-services/index.html

Contents

ORACLE®

ORACLE®

1 Preface

1.1 Introduction

This guide provides guidance to users for the issues within the application. It describes various methods to figure out the error and then troubleshoot it.

1.2 Audience

This guide is intended for the software developers and software testers.

1.3 Document Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

1.4 List of Topics

This guide is organized into the following topics. The topic 1 to 3 are generic and applicable for Oracle Banking Microservice Architecture platform wide troubleshooting and topic 4 to 7 are specific to Oracle FLEXCUBE Onboarding product.

Table 1: List of Topics

Topic	Description
Preface	This topic provides the information about this guide and intended audience. It also lists the various chapters covered in this User Manual.
Troubleshooting Technical Flows	This topic describes the platform wide troubleshooting of technical flows. This includes UI side checking, Service side logs, tracing using zipkin, debugging using ELK stack and some environment issues of weblogic
Health Checks	This topic describes the health check measures and observability required.
Troubleshooting Functional Workflows	This topic describes Oracle FLEXCUBE Onboarding specific troubleshooting of flows and specific data stores that can be checked for issues.
Business Error Codes	This topic provides the list of error codes expected out of Oracle FLEXCUBE Onboarding.

2   Troubleshooting Technical Flows

This chapter describes various programming issues, possible causes, and solutions to resolve the issues. This topic contains the following sections:

2.1   Where is the problem

Troubleshooting the problem in distributed system could be bit challenging if not understood fully. Each product has UI application components and service side application components. Each side requires different troubleshooting techniques and various logs that can be used to corroborate the problem.

The below picture shows that on service side, it is important to establish the area of the problem. This can be achieved by complete understanding of UI, Service side flows along with the data architecture of application.

Figure 1: Distributed Flow across Micro Services

ORACLE®
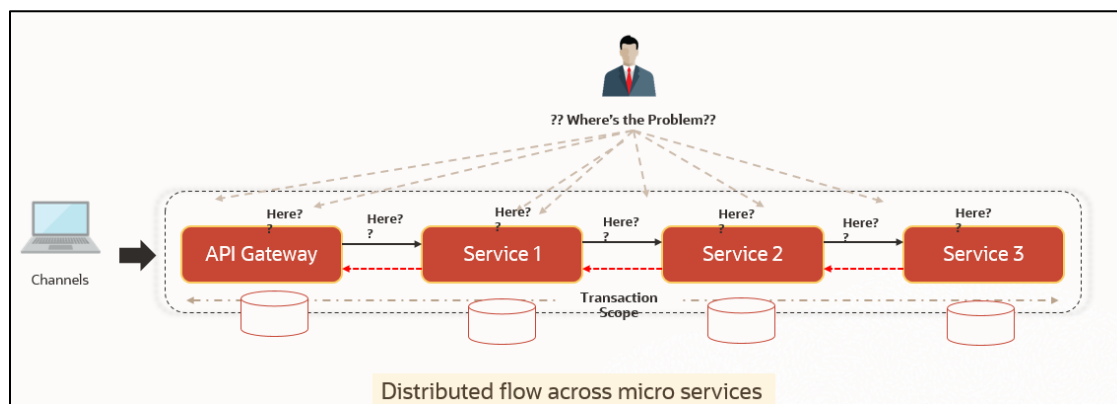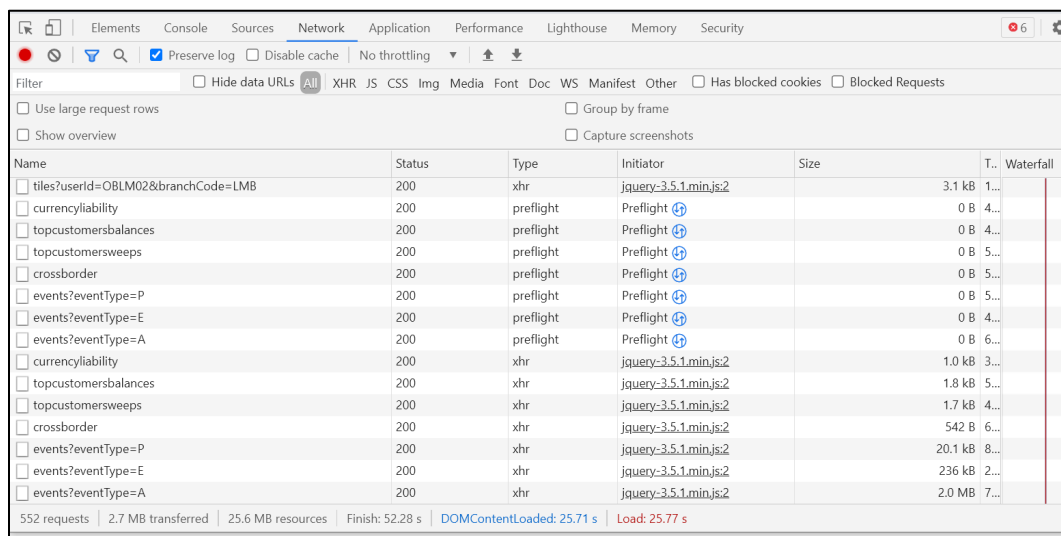
2.2 Preliminary checks from UI

This section describes how to launch the application and check for the basic errors. Perform the following steps:

1. Launch the application with delegated URL.

2. Press **F12** key, and select **Inspect and See network** tab.

3. Verify that all the calls responses are successful. Usually RED color indicates non 2xx HTTP response.

Figure 2: Call Responses

Figure 3: Non 2xx response

Figure 4: Details of Non 2xx Response
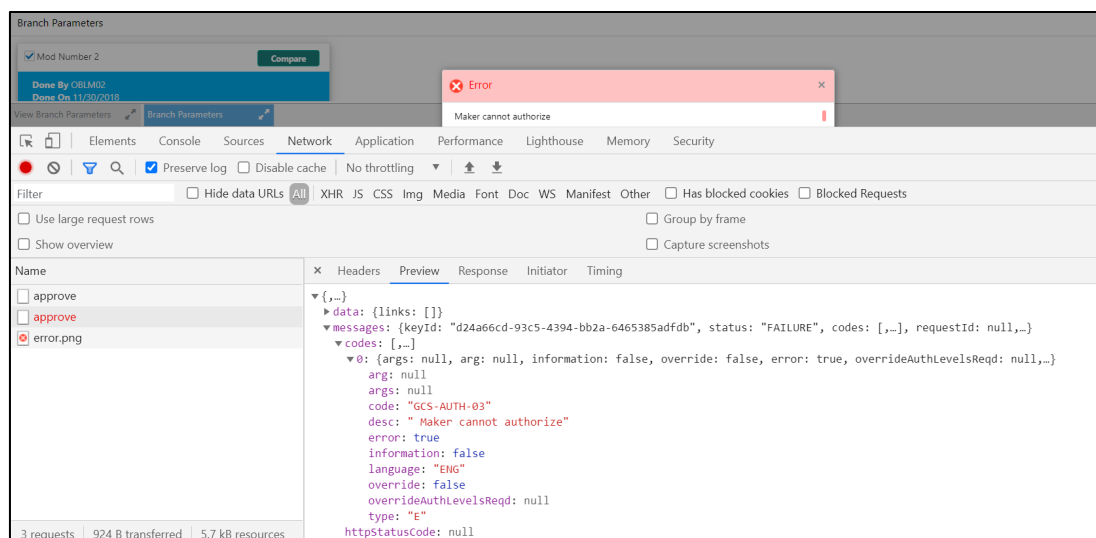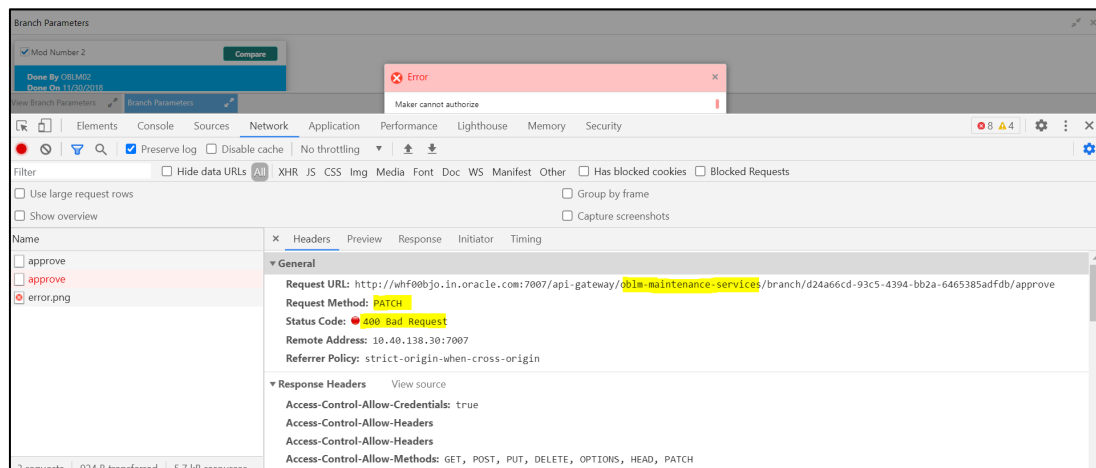
4. You can also export the trace using export option in browsers.

    Example in Chrome you would see this option as below

Figure 5: Export Option
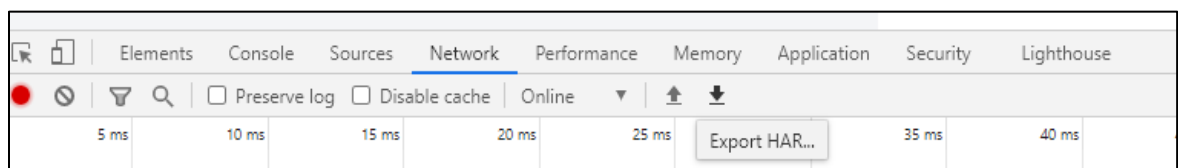
NOTE: You can also use tools such as **fiddler** and **wireshark** to get the browser to API gateway web traffic. This would help to investigate exact request and response payloads exchanged between UI and API Gateway.

ORACLE®

2.3 Preliminary checks from Service Log files

Each micro service sub domain war deployments can generate the log files in weblogic server. The configuration of this log can be found at logback.xml

```
<root level="INFO">
    <appender-ref ref="FILE" />
</root>
```

In production scenarios, it is wise to ensure the root level is configured as ERROR so that log files doesn't get overwhelmed

Refer Weblogic documentation to know the path where these files are generated. In on-premises cases, these log files can be zipped and send for remote troubleshooting purposes.

ORACLE®

2.4 Troubleshooting using Zipkin Traces

2.4.1 Setting Zipkin Server
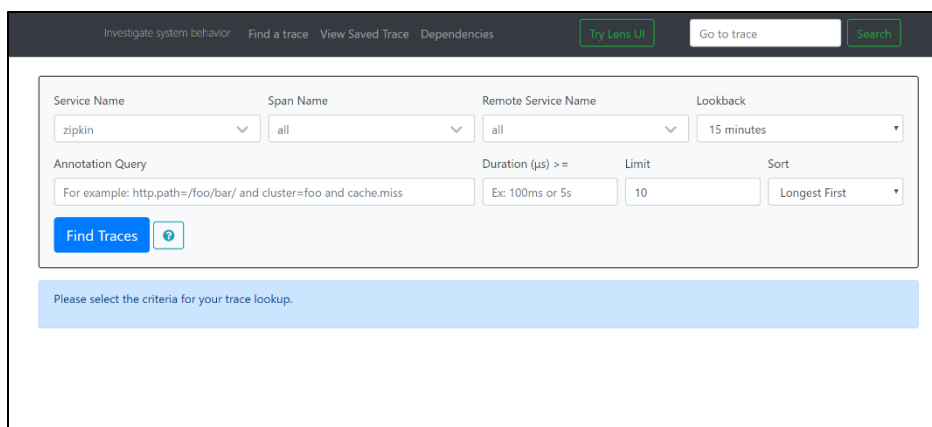
Refer below document for installation document for Zipkin installation

https://docs.oracle.com/cd/F41347_01/PDF/Installation_Guide/ANNEXURE-2.pdf

2.4.2 Login to Zipkin

The basic layout of Zipkin looks as follows

Figure 6: Layout of Zipkin
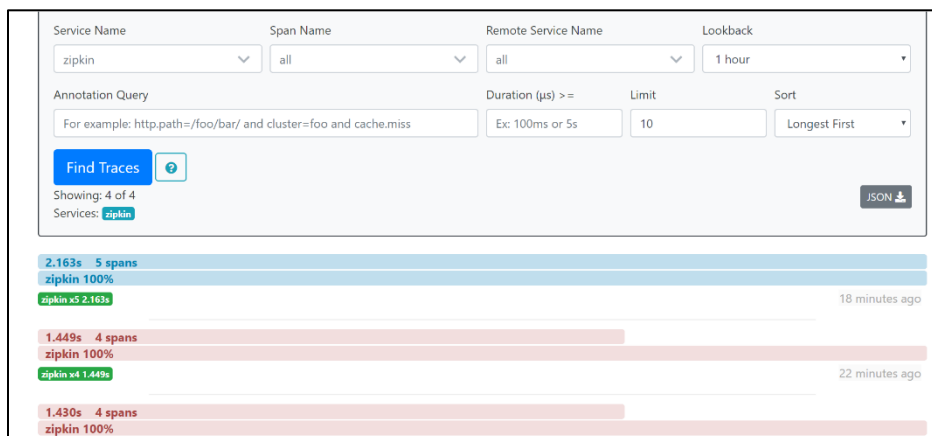
We can find the traces of required API calls and services using the above search options given in the user interface. The search options given in the user interface are self-explanatory and there is another UI option (Try Lens UI). It's given a different user interface with the same functionality.

Figure 7: List of Traces

The list of the traces can be seen as the above screen. I made some error API calls to showcase how to track errors. The blue listings show the successful API hits and the red listings indicate errors. Each block indicates a single trace in the listings.

Opening an individual trace shows the below-shown screen.

Figure 8: Individual Trace
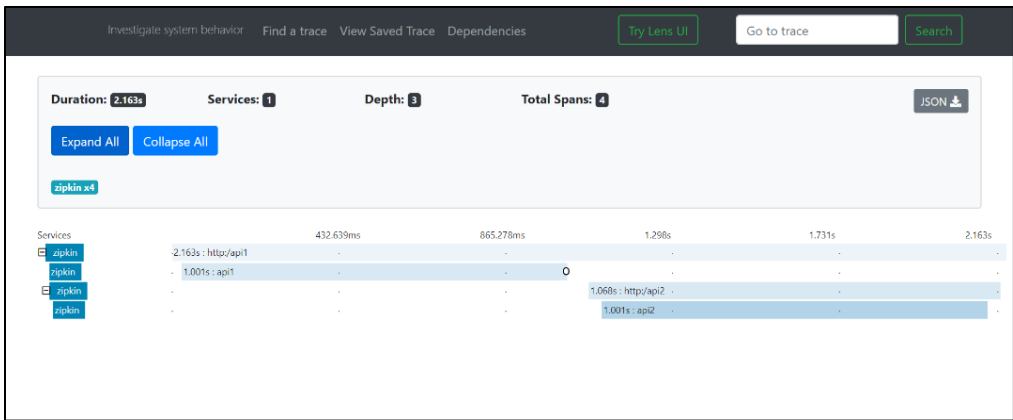
The above-shown image describes the time taken for each block. As I created 2 custom spans inside 2 service calls, we can find a total of 4 blocks. The time taken for an individual block can be seen above. Clicking an individual block shows the following details.

Figure 9: Details of Individual Block
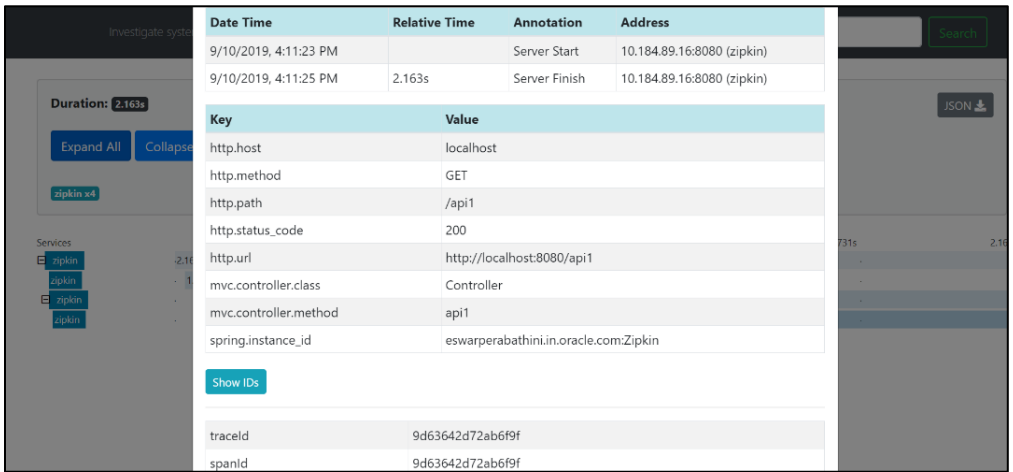
The details of the specific span block are shown above and the logging events can also be seen in the Zipkin UI as small circular blocks. An example of an error log is shown below.

Figure 10: Sample Error Log
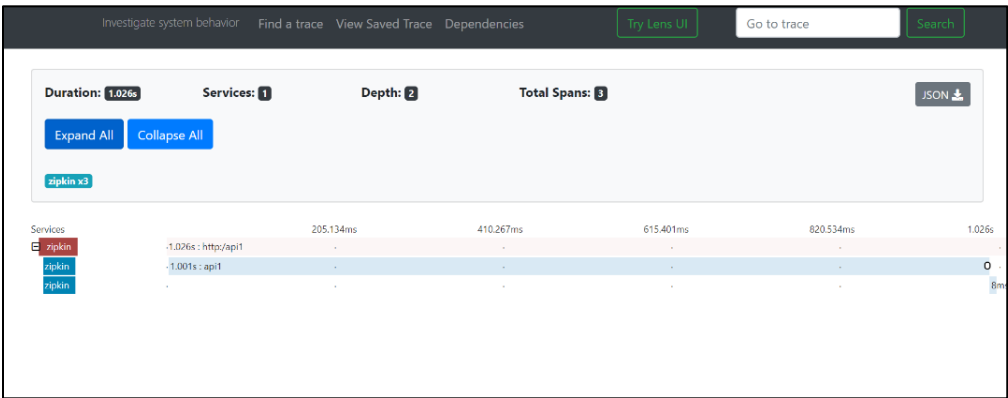
Clicking on the error portion gives a clear detail about the error and where the error has arisen. An example is shown below.

Figure 11: Details of Error
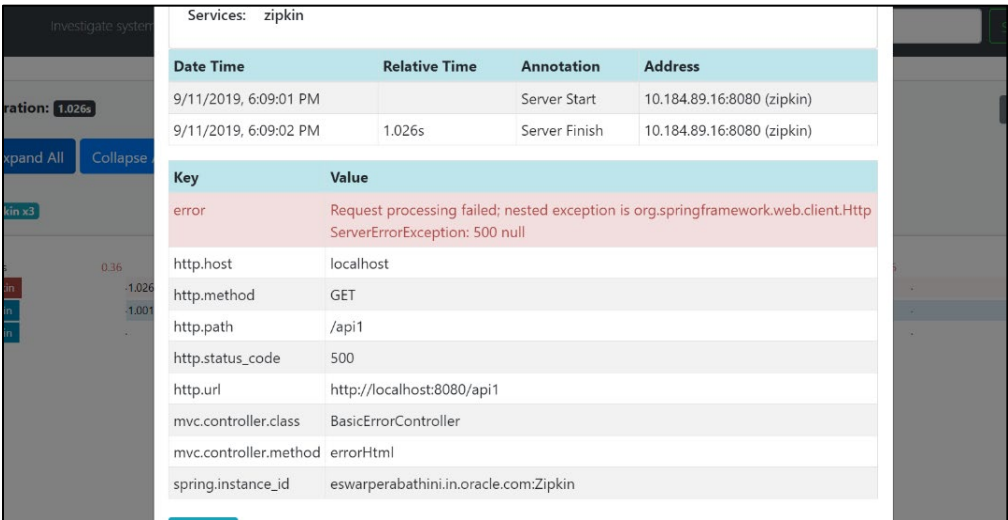
If the Lens UI is used in Zipkin, the above screenshots are not applicable but are relatable to the Lens UI as well.

Traces of the application can be found using TraceId, which can be found in the debug logs of the deployment when spring-cloud-sleuth is included in the dependencies (Included in spring-cloud-starter-zipkin dependency). Clicking the dependency tab gives the dependency graph info between micro-services. An example dependency graph is shown below.

Figure 12: Sample Dependency Graph
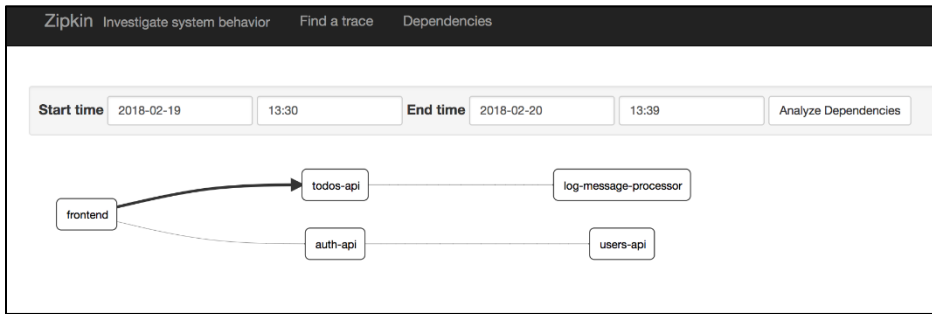
2.4.3 Zipkin Issues

2.4.3.1 Application service not registered

Perform the following steps to find the cause of this error:

1. Check the applications, which are sending the trace report to zipkin server from **Service Name** drop-down list.

Figure 13: Find Traces
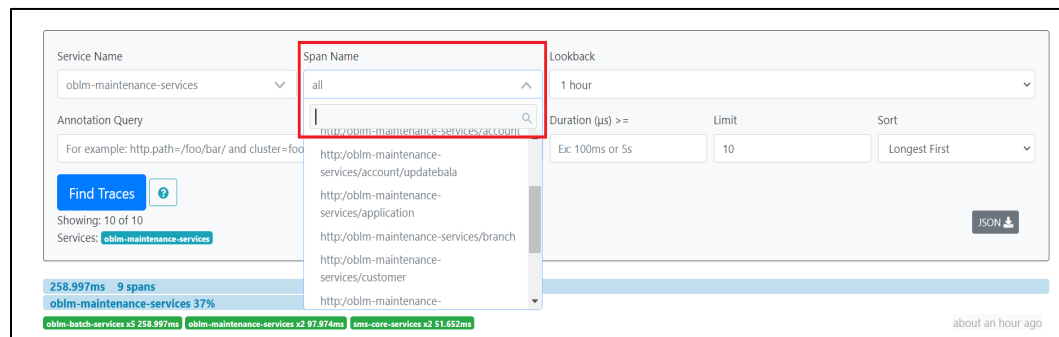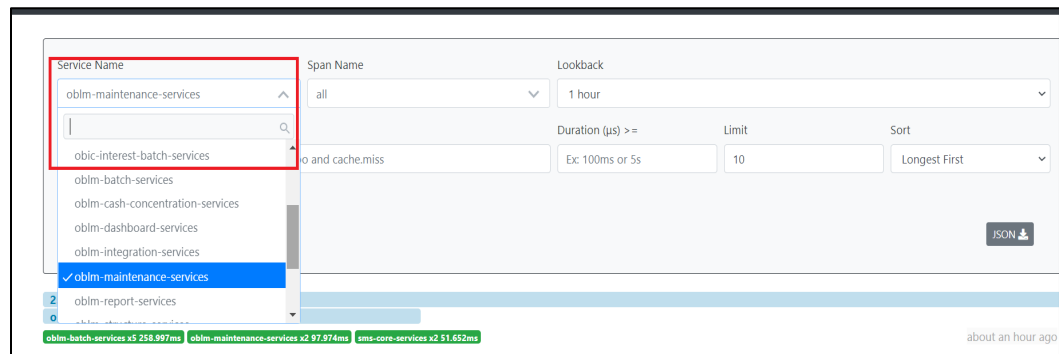
ORACLE®

2. If the required application is not listed in Zipkins, check the application.yml file for zipkin base URL configuration. The shipped application.yml should have the zipkin entry.

Figure 14: Application.yml File

```
x logback.xml    ☑ application.yml ⊠
 1# Copyright (c) 2016, Oracle. All rights reserved.
 2 spring:
 3   application:
 4     name: oblm-maintenance-services
 5   main:
 6     allow-bean-definition-overriding: true
 7   autoconfigure:
 8     exclude: org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration, org.springframewo
 9   zipkin:
10     baseUrl: ${plato.services.zipkin.url:http://localhost:9411}
11     service:
12       name: ${spring.application.name}
13     sender:
14       type: ${plato.services.zipkin.sender.type:web}
15   sleuth:
16     sampler:
17       percentage: ${plato.services.sleuth.sampler.percentage:0.1}
18
19 eureka:
20   instance:
21     prefer-ip-address: true
22     securePortEnabled: ${isSslEnabled:false}
23   client:
24     registerWithEureka: true
25     fetchRegistry: true
26 #   preferSameZone: true
```

NOTE: Every service should have **spring-cloud-sleuth-zipkin** dependency added in build gradle file for the service to generate and send trace Id and span Id.

Compile group: 'org.springframework.cloud', name: 'spring-cloud-sleuth-zipkin', version: '2.1.2.RELEASE'.

Figure 15: Branch Common Services

ORACLE®
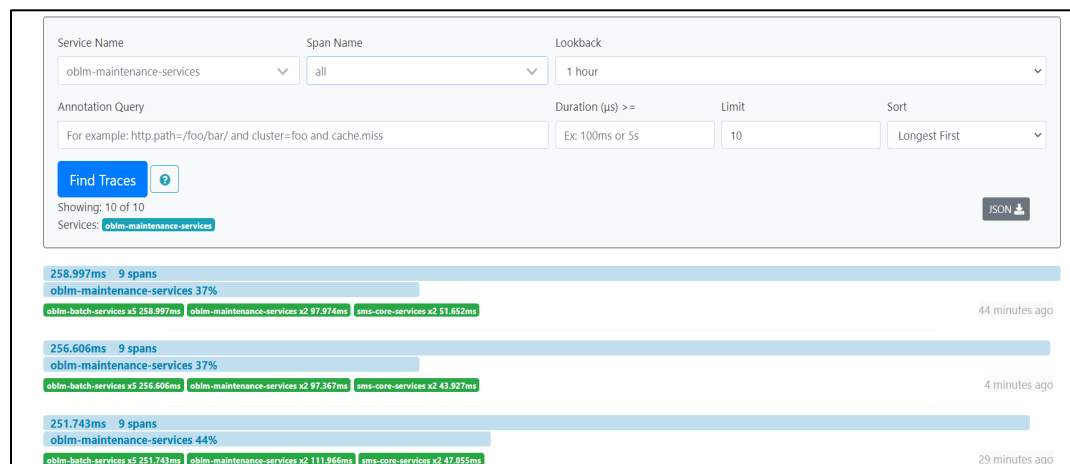
Figure 16: Branch Common Services Trace
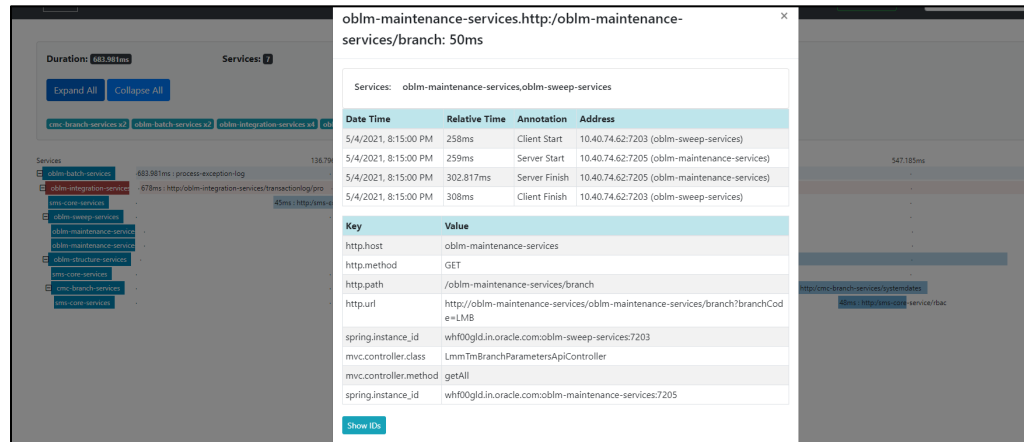
2.4.3.2 404 error

If there is 404 error, check if the zipkin-server.jar is running in the system where the application is deployed.

To check this execute the following command

netstat -ltnup | grep ':9411'

output should be like:

tcp6   0   0 :::9411   :::*   LISTEN   10892/java

Here 10892 is the PID.

2.4.3.3 Unable to change zipkin default port number

Zipkin default port number is not editable. Hence, verify that the port 9411 is available to start Zipkin-server.jar file. Also check if the zipkin url is provided for the services in the PROPERTIES table.

SELECT * FROM PROPERTIES WHERE KEY LIKE '%zipkin%'

Figure 17: ELK Setup
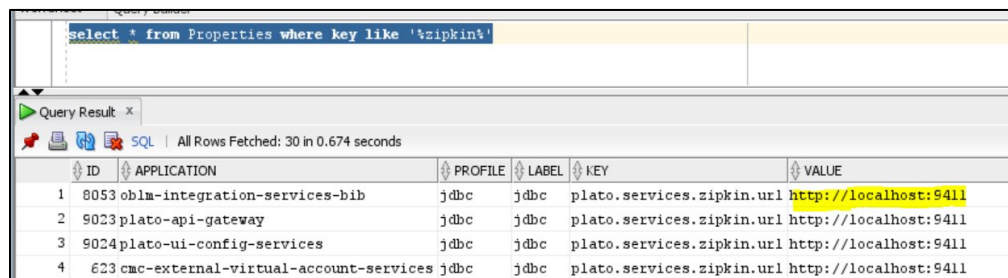
2.5 Troubleshooting Logs using ELK stack

2.5.1 Setting up ELK

Download the Elastic search from https://www.elastic.co/downloads/elasticsearch

Kibana download https://www.elastic.co/downloads/kibana

Logstash download https://www.elastic.co/downloads/logstash

Default port for Elastic search- 9200

Default port for Kibana- 5601

Step to run ELK:

Run the elasticsearch.sh file present inside /scratch/software/ELK/elasticsearch-6.5.1/bin

- Edit network.host to "localhost" and port if necessary. This should be enough for it to run.

- Start: **nohup bin/elasticsearch &**

Configure Kibana to point the running instance of elastic search in kibana.yml file as below.

Figure 18: Logstash Configuration

```
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "whf00peb"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# This setting was effectively always `false` before Kibana 6.3 and will
# default to `true` starting in Kibana 7.0.
#server.rewriteBasePath: false

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name.  This is used for display purposes.
#server.name: "your-hostname"

# The URL of the Elasticsearch instance to use for all your queries.
elasticsearch.url: "http://localhost:9200"

# When this setting's value is true Kibana uses the hostname specified in the server.host
```

Configuring Logstash consists of 3 steps:

1. **Input**- This configuration is required to provide the log file location for the Logstash to read from.

2. **Filter**- Filters in logstash is basically used to control or format the read operation(Line by line or Bulk read)

ORACLE®

3. **Output**- In this section we provide the running elastic search instance to send the data for persisting.

Figure 19: Kibana

```
logstash.conf

#Point to the application logs
input {
  beats {
    port => 5044
  }
}
#Provide the parsing logic to transform logs into JSON
filter {
 # Adding @metadata needed for index sharding to Filebeat logs
  mutate {
    copy => {
      "[fields][app_name]" => "[@metadata][app_name]"
      "[fields][env]" => "[@metadata][envt]"
    }
  }

  #If log line contains tab character followed by 'at' then we will tag that entry as stacktrace
  if [message] =~ "\tat" {
    grok {
      match => ["message", "^(\tat)"]
      add_tag => ["stacktrace"]
    }
  }
```

```
  #Grokking Spring Boot's default log format
  grok {
    match => [ "message", "%{TIMESTAMP_ISO8601:timestamp}\s+%{LOGLEVEL:severity}\s+\[%{DATA:service},%{DATA:trace},%{DATA:span}
  }

  #Parsing out timestamps which are in timestamp field thanks to previous grok section
  date {
    match => [ "timestamp" , "yyyy-MM-dd HH:mm:ss.SSS" ]
  }
  fingerprint {
    source => "message"
    target => "[@metadata][fingerprint]"
    method => "MD5"
    key => "test"
  }

  ruby {
    code => "event.set('[@metadata][prefix]', event.get('@timestamp').to_i.to_s(16))"
  }
}

#Ingest logs to Elasticsearch
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "%{[@metadata][app_name]}-%{[@metadata][envt]}-%{+YYYY.MM.dd}"
    document_id => "%{[@metadata][prefix]}%{[@metadata][fingerprint]}"
  }
  stdout { codec => rubydebug }
}
```

2.5.2 Accessing Kibana

- Goto path ../kibana-7.8.1-linux-x86_64/config/kibana.yml

- Edit server.host: "0.0.0.0" for access outside host and server.port: <any port, defaults to 5601>

- Validate elasticsearch properties - it defaults to localhost:9200

- Goto : http://host:port you should be able to see the Kibana console UI. Kibana needs elasticsearch to be UP as it creates indexes & fetches logs from it.

- Start: nohup bin/kibana &

Figure 20: Accessing Kibana
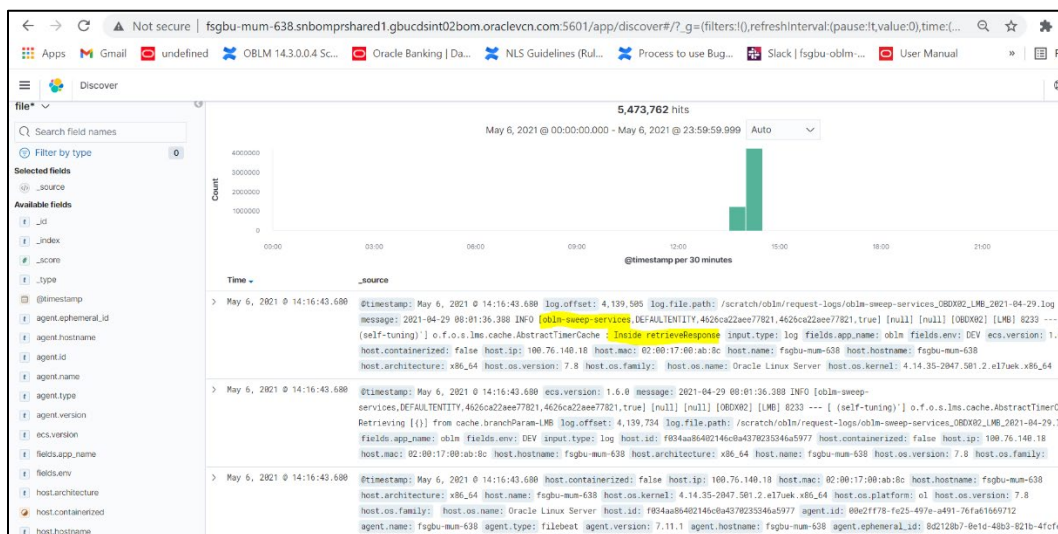
2.6   Troubleshooting Logs Kafka issues

2.6.1 Setting up kafka

Install Kafka and Zookeeper which is shipped along with the OSDC

Now first we need to start zookeeper and then kafka. Please flow steps mentioned here

https://kafka.apache.org/quickstart

2.6.2 Check if kafka is running

Run cmd $ netstat –tlnp | grep :9092

   (9092 is default port of kafka)

Possible issue while starting kafka

1. Kafka is not starting may be because zookeeper is not yet started

    run cmd $ netstat –tlnp | grep :2181

   (2181 is default port of zookeeper)

    if any services is not running on this port means zookeeper is down

2. Check if any permission issue is there for kafka log folder.

Create console producer and consumer for troubleshooting

here is a use full reference for that http://cloudurable.com/blog/kafka-tutorial-kafka-from-command-line/index.html

2.6.3 Some references that can be useful

https://docs.cloudera.com/documentation/kafka/latest/topics/kafka_faq.html

2.7 Troubleshooting Environmental Issues

This section contains the following subsections:

- 2.7.1 Possible issues while deploying services
- 2.7.2 Possible issues in login and screen launch

2.7.1 Possible issues while deploying services

This subsection describes the possible issues that may occur in the environment.

2.7.1.1 Service deployment is failing due to flyway

If the service deployment is failing due to flyway, verify that the object or record is already present and make changes in the flyway scripts accordingly.

You may check **flyway_schema_history** table of the respective schema for finding the flyway script entries.

2.7.1.2 Other possible issues

The other possible issue while deploying services could be multiple version of dependency jars present in the war file. For example,
"weblogic.application.naming.EnvironmentException: duplicate persistence units with name PLATO in scope cmc-customer-services-5.3.0.war."

2.7.2 Possible issues in login and screen launch

This subsection describes the possible issues that may occur while logging in and launching the screens.

2.7.2.1 Login page is not launching

If the login page is not launching, check if the app-shell war file is deployed. If it is deployed, make sure that the war file is up and running in the deployed managed server and try to login. In addition, check if you are logged in with the appshell URL according to the war file deployed.

Also, check if the required component-server wars like cmc-component-server, oboflo-component-server etc are also deployed along with the app-shell.

For example, http://whf00bjp.in.oracle.com:7403/app-shell/index.jsp will load the login page of the application.

ORACLE®

Figure 21: Login Page

2.7.2.2 Unable to login after launching the application

If you are not able to login after the application is launched, make sure that the plato-api-gateway service, plato-ui-config service, sms-core-service, and common-core-services are up and running.

Figure 22: Services

PLATO-API-GATEWAY	n/a (1)	(1)	UP (1) - whf00bjo.in.oracle.com:plato-api-gateway:7007
PLATO-BATCH-SERVER	n/a (1)	(1)	UP (1) - whf00bjo.in.oracle.com:plato-batch-server:7013
PLATO-DISCOVERY-SERVICE	n/a (1)	(1)	UP (1) - whf00bjo.in.oracle.com:plato-discovery-service:7003
PLATO-O	n/a (1)	(1)	UP (1) - whf00bjo
PLATO-ORCH-SERVICE	n/a (1)	(1)	UP (1) - whf00bjo.in.oracle.com:plato-orch-service:7017
PLATO-UI-CONFIG-SERVICES	n/a (1)	(1)	UP (1) - whf00bjo.in.oracle.com:plato-ui-config-services:7009
SMS-CORE-SERVICES	n/a (1)	(1)	UP (1) - whf00bjo.in.oracle.com:sms-core-services:7011

2.7.2.3 Unable to login after restarting the services

If you are not able to login after restarting the services, make sure that the LDAP server is up and running, and check if the entered credentials are correct.

2.7.2.4 Retail Banking menus are not displayed after logging in

After you log in, if the Retail Banking menus are not displayed, map the functional activity codes in the table SMS_TM_ROLE_ACTIVITY. Once it is mapped, check if the corresponding role is assigned to your user id.

2.7.2.5 Screens are not launching after logging in

If you are not able to launch the screens after logging in, make sure that the respective services are up and running.

NOTE: Verify the VPN connection while trying to troubleshoot the issues related to page launching, etc.

ORACLE®

3   Health Checks

Until we get heath check APIs implemented, the health need to be monitored using weblogic JVM managed server status and Eureka instance.

Figure 23: Health Checks

| Application	AMIs	Availability Zones	Status
CMC-ACCOUNT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-account-services:7005
CMC-ADVICE-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-advice-services:7005
CMC-BASE-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-base-services:7005
CMC-BRANCH-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-branch-services:7005
CMC-BUSINESSOVERRIDES-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-businessoverrides-services:7005
CMC-CHECKLIST-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-checklist-services:7005
CMC-COMMENTS-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-comments-services:7005
CMC-CURRENCY-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-currency-services:7005
CMC-CUSTOMER-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-customer-services:7005
CMC-DATASEGMENT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-datasegment-services:7005
CMC-DOCUMENT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-document-services:7005
CMC-EXTERNAL-CHART-ACCOUNT-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-external-chart-account-services:7005
CMC-OBCBS-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-obcbs-services:7005
CMC-OBRH-SERVICES	n/a (1)	(1)	UP (1) - whf00cdl.in.oracle.com:cmc-obrh-services:7005

3.1 Weblogic

3.1.1 Unable to login to weblogic console

Sometimes you are unable to login to weblogic console or console is down when trying to deploy or re-deploy services

This issue can be fixed by restarting weblogic domain from server. Perform the following steps:

1. To stop the weblogic server, which is already running, go to path "/Oracle_Home/user_projects/domains/bin" and execute **sh** file with "**./**" prefixing to it. For example, **./stopWebLogic.sh**.

2. Once the server is stopped, try to start the server by executing below command, so that it can run in the background. For example, **./startWebLogic.sh &**.

Figure 24: Error Message
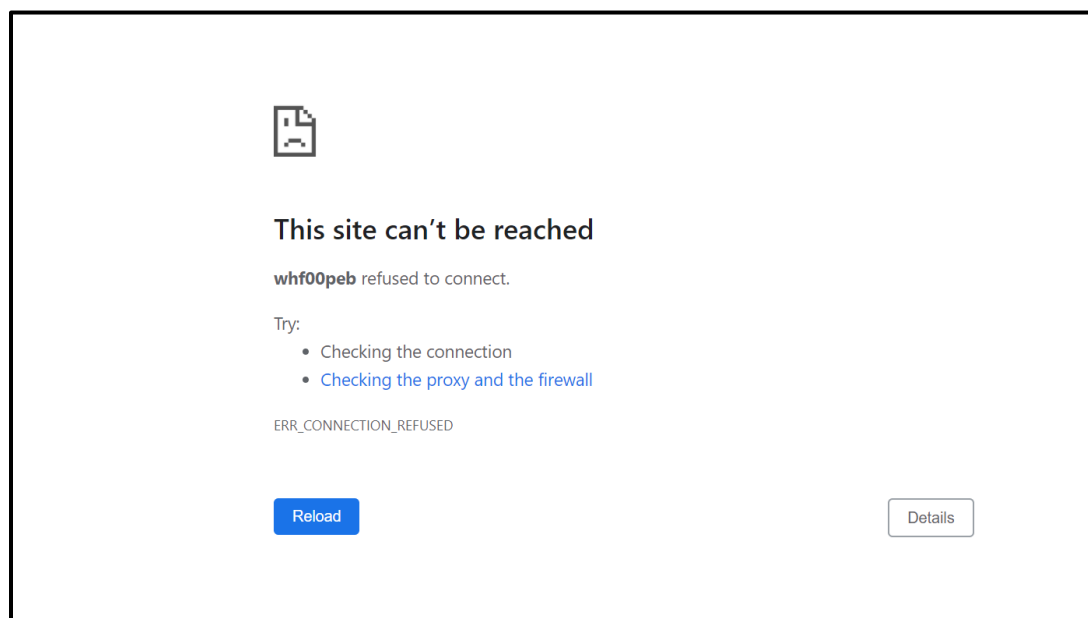
3.1.2 Unable to stop a service

If you are not able to stop a service, which is already running, try to bring down the managed server and remove the war file.

3.1.3 GC Overhead limit exceeded/OutOfMemoryException error

If there is an error like GC Overhead limit exceeded or OutOfMemoryException is thrown while starting the services, the following details need to be shared.
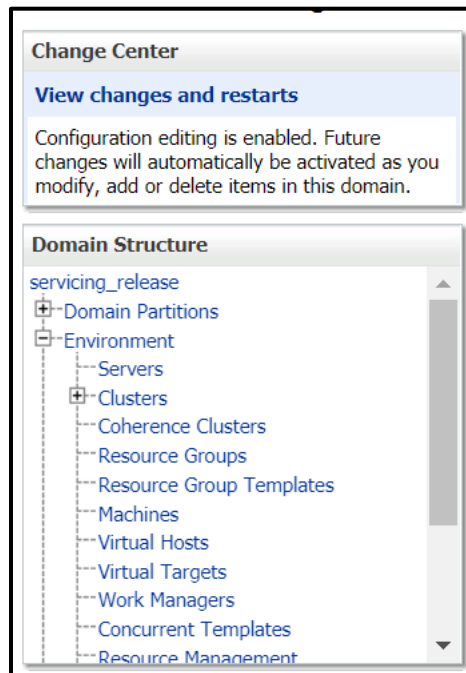
- Heap Dump

- Configuration of environment

For quick fix, try to restart the managed server or try to increase the memory allocated to the managed server.

Perform the following steps to increase the memory:
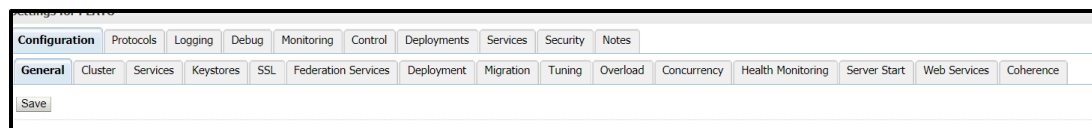
1. Click **Servers**.
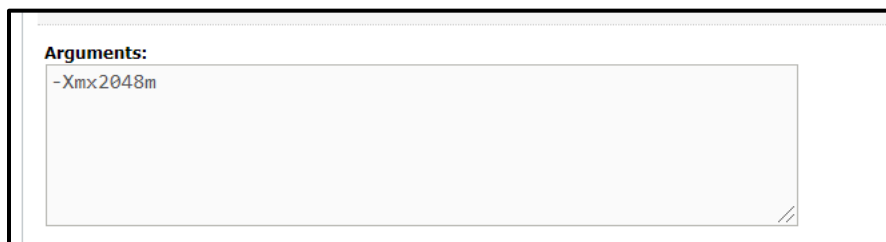
Figure 25: Domain Structure

2. Select the managed server from which you are getting OutOfMemoryException or GC Overhead Limit exceeded, and click **Server Start** tab.

Figure 26: Managed Servers

ORACLE®

3. Specify the memory (which needs to be increased) according to requirement in 512, 1024, 2048 etc., and restart the managed server to fix the issue.
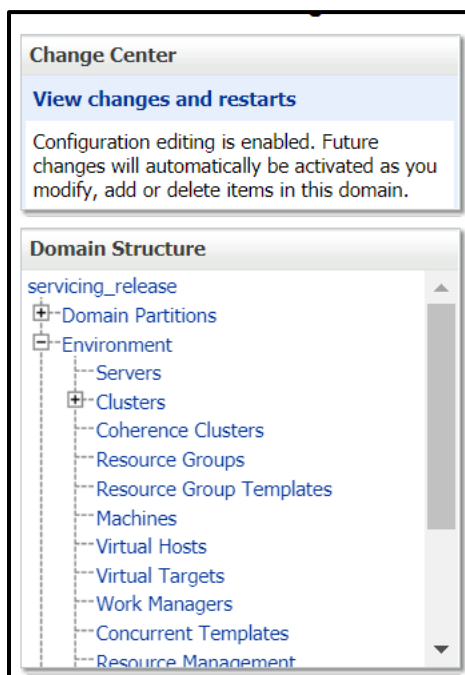
Figure 27: Memory Update

3.1.4 Managed server is failed or not reachable

If the managed server is in failed or not reachable state, perform the following steps to restart the managed server:

1. Go to **Servers**, select **Control** tab, and then select the managed server.

Figure 28: Domain Structure

ORACLE®

2. Click **Shutdown**.

Figure 29: Control Tab
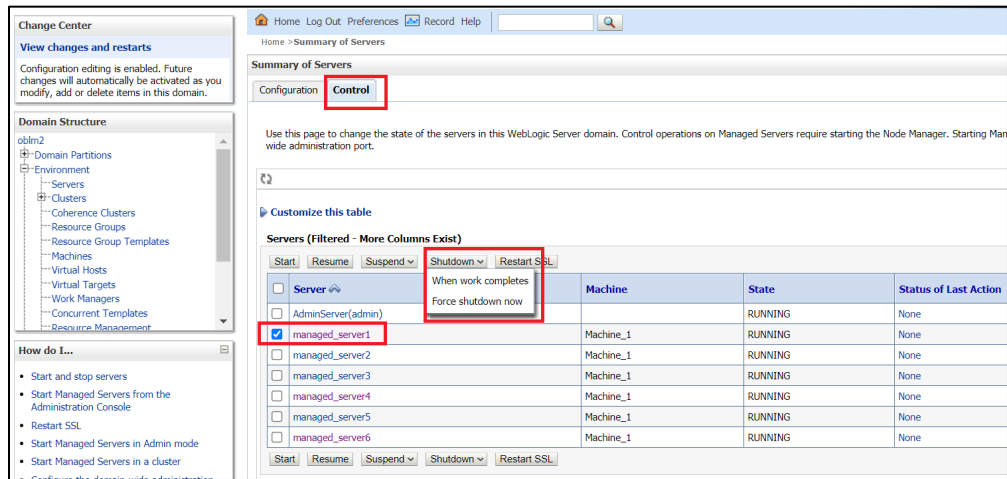
3. After you bring down the server, click **Start** to restart the server.

3.1.5 weblogic.application.ModuleException error

If there is an error like "weblogic.application.ModuleException: Context path '/obremo-rpm-maintanence-services' is already in use by the module", make sure that the re-deploying service is removed properly. If the issue persists, try to restart the managed server.

ORACLE®

3.2 Database

3.2.1 Configure Data Sources in WebLogic

To add the Data Sources, follow the below steps:

1. Go to **Services**, select **Data Sources** option.

Figure 30: Data Sources
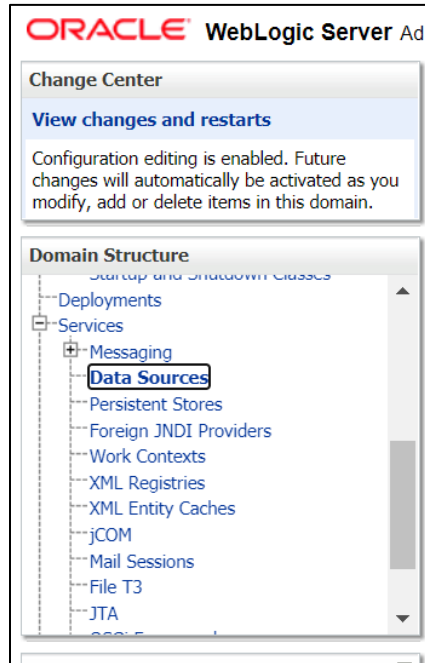
2. Click on **New** button and add the data source providing the required details.

Figure 31: Summary of JDBC Data Sources

ORACLE®

Figure 32: Create a New JDBC Data Source

3.2.2 Day0 Scripts

Make sure that all the Day0 scripts available in the product package are run successfully.

ORACLE®

4 Troubleshooting Application Workflows
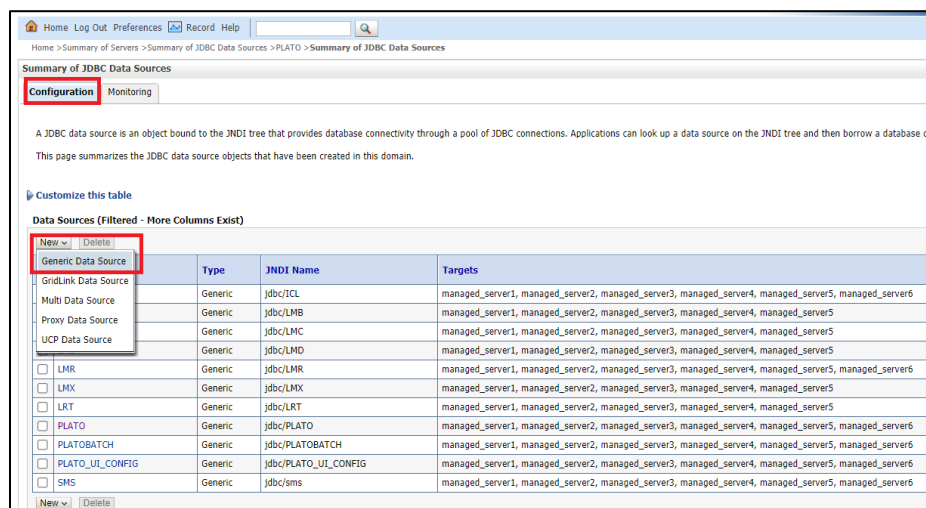
On successful login, the Oracle FLEXCUBE Onboarding dashboard screen appears depending on the user privileges.

Figure 33: Oracle FLEXCUBE Onboarding Dashboard
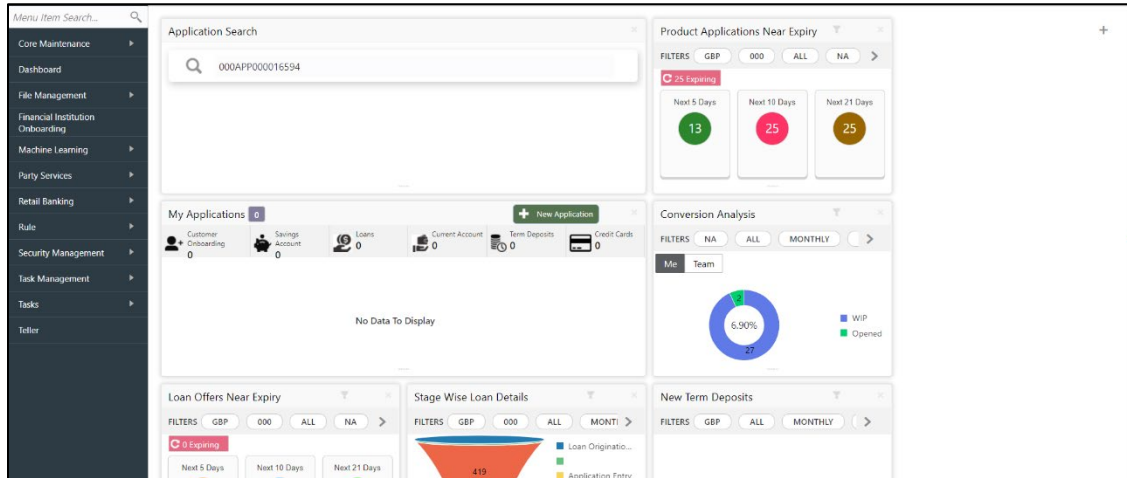
- **User Role Issues**

  Role Profile includes access rights to the functional activities that are common to a group of users. A user can be linked to a Role Profile by which you give the user access rights to all the functional activities in the Role Profile.

  Go to: **Security Management** > **Role** screen

  Figure 34: Role Maintenance

  Only authorized users can access the system with the help of a unique User Login ID and password. The user profile of a user contains the details of the user in four sections - User details, Status, Other details and User role branches.

Go to: **Security Management** > **User screen**

Figure 35: Users Maintenance

NOTE: Make sure that the required Role and User Applications are mapped to the user.

4.1 First level issues

This subsection describes the possible issues that may occur during the basic investigation.

4.1.1 Error Message not shown

If there is any improper calls, check ERTB_MSGS table of the respective schema to understand the cause of the error. Open the Networks tab (F12), check the error code in the response.

SELECT * FROM ERTB_MSGS WHERE ERR_CODE='GCS_AUTH-03'

Figure 36: Error Message not shown
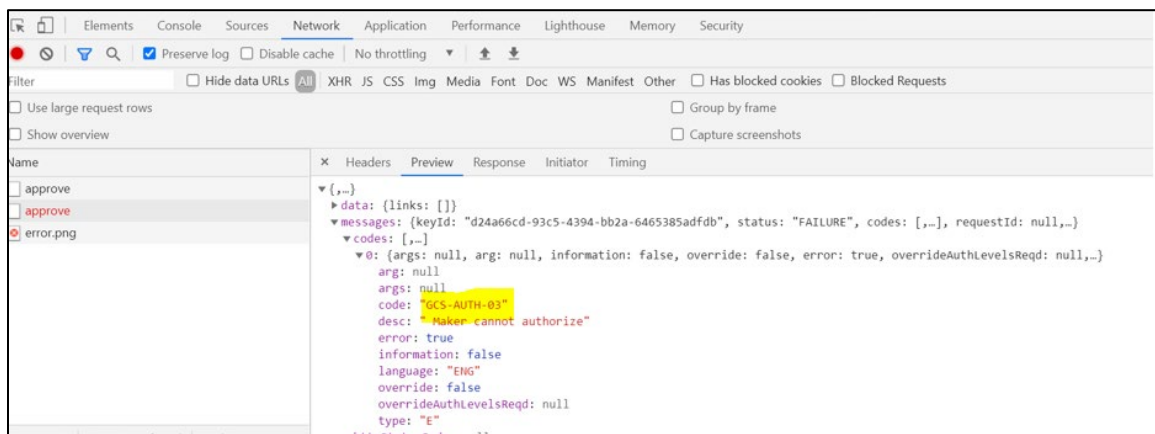
4.1.2 Setting Log file path

Log generation path needs to be defined in **PLATO_LOGGER_PARAM_CONFIG** table of PLATO schema.

select * from PLATO_LOGGER_PARAM_CONFIG;

Figure 37: Setting Log file path

ORACLE®
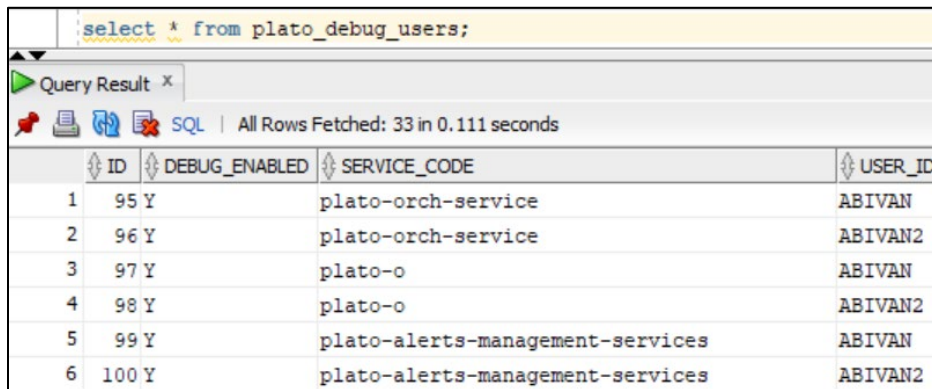
4.1.3 Dynamic log generation issues

For generating dynamic service logs, you need to insert data to **PLATO_DEBUG_USERS** table as shown in below screenshot.

select * from PLATO_DEBUG_USERS;

Figure 38: Dynamic log generation issues
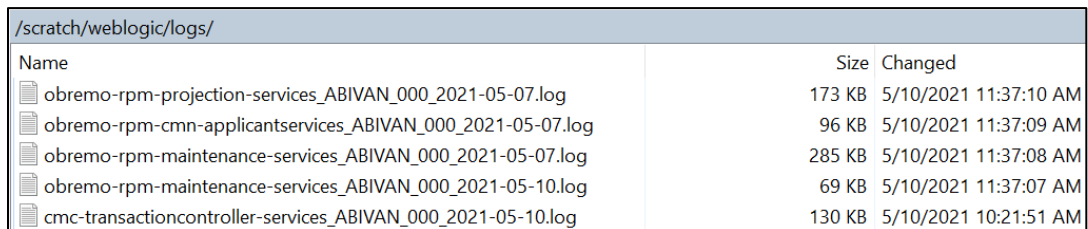
	ID	DEBUG_ENABLED	SERVICE_CODE	USER_ID
1	95	Y	plato-orch-service	ABIVAN
2	96	Y	plato-orch-service	ABIVAN2
3	97	Y	plato-o	ABIVAN
4	98	Y	plato-o	ABIVAN2
5	99	Y	plato-alerts-management-services	ABIVAN
6	100	Y	plato-alerts-management-services	ABIVAN2

*select * from plato_debug_users; — Query Result — All Rows Fetched: 33 in 0.111 seconds*

NOTE: Login to WINSCP and check server logs. Log files for each service will be generated based on the user_id, branch_code and date at the path provided in the plato_logger_param_config table.

Figure 39: Server logs

/scratch/weblogic/logs/

Name	Size	Changed
obremo-rpm-projection-services_ABIVAN_000_2021-05-07.log	173 KB	5/10/2021 11:37:10 AM
obremo-rpm-cmn-applicantservices_ABIVAN_000_2021-05-07.log	96 KB	5/10/2021 11:37:09 AM
obremo-rpm-maintenance-services_ABIVAN_000_2021-05-07.log	285 KB	5/10/2021 11:37:08 AM
obremo-rpm-maintenance-services_ABIVAN_000_2021-05-10.log	69 KB	5/10/2021 11:37:07 AM
cmc-transactioncontroller-services_ABIVAN_000_2021-05-10.log	130 KB	5/10/2021 10:21:51 AM

ORACLE®

4.1.4 Call is failing in Gateway

If any API call is failing in Gateway, hit the same API endpoint without passing through api-gateway via postman.

Figure 40: Call is failing in Gateway
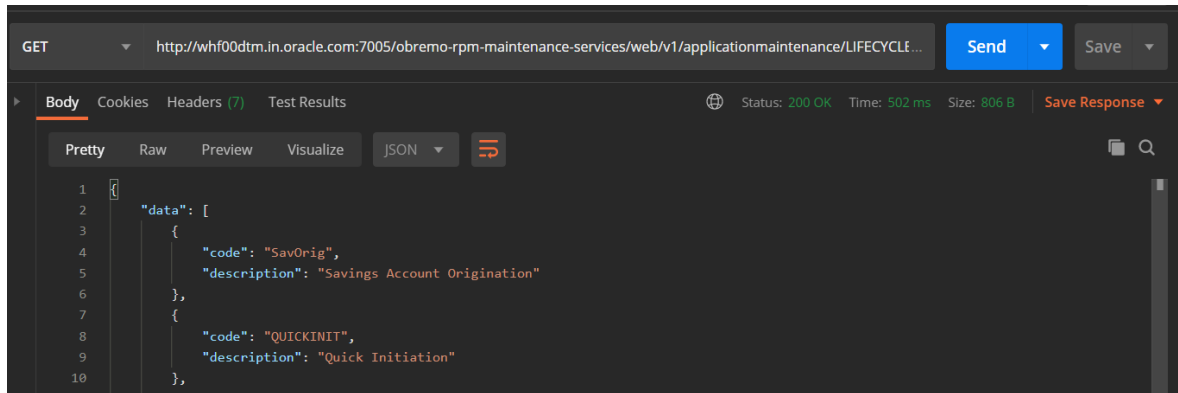
NOTE: Restart the specific services if required.

4.1.5 Code error in GCS side

If there is any error in GCS side codes, use java de-complier to debug the error.

4.1.6 404 error

The possible causes for 404 error are as follows:

- Check service is not running on Eureka

- Check if service is deployed in Weblogic

4.1.7 500 internal error

The possible causes for 500 internal error are as follows:

- Issue with Oracle Banking Microservices Architecture entries

- Issue with Eureka

- Service may not be up

- Issue with any peace of code

The server-side debugging is needed for the above-mentioned issues, if it is not captured in logs.

4.2 Transaction data verification

Follow the best practices mentioned below to avoid getting any errors:

- In the **IN** request and **OUT** response, verify that all the field data is going to service side.

- If there is any error related to SMS, check for the availability of SMS entries.

- Validate the endpoints and data.

- Validate the request headers passed during the API call.

- Verify that the data entered in the screen is accurate.

4.2.1 "Apply Now" is Failing in Product Catalogue

If "Apply Now" in Product Catalogue is failing, troubleshoot using the below points

- Check if conductor war and plato-orch-service war is deployed in Weblogic

- Check whether "PLATO-O" and "PLATO-ORCH-SERVICE" is registered in Eureka

PLATO-O	n/a (1)	(1)	UP (1) - plato-o:8001
PLATO-ORCH-SERVICE	n/a (1)	(1)	UP (1) - whf00dtm.in.oracle.com:plato-orch-service:7011

- Check whether the INITIATION workflow DSL is imported.

    - Front-End Menu: **Tasks → Business Process Maintenance → Search** for INITIATION workflow

- Check whether obremo-rpm-projection-services is up and running as this service is required during INITIATION(Apply Now)

- Check whether Sequence Generator service is up and running

SEQUENCEGENERATORSERVICE	n/a (1)	(1)	UP (1) - whf00dtm.in.oracle.com:sequencegeneratorservice:7020

- Refer **2.2 Preliminary Check for UI** section to see if any API call is failing

ORACLE®

4.3 Party Module Integration Troubleshooting

This section describes the possible issues that may occur in Party Module integration. The possible issues and causes are described in the following subsections:

4.3.1 Existing Customer Details Fetch is failing

If in Customer Information data-segment, the existing customer details is not fetching, follow the below steps to

1. Refer **2.2 Preliminary Check** for UI to see if any Party API is failing

2. Check Oracle Banking Routing Hub Audit Request to see if any Oracle Banking Routing Hub calls to Party Module has failed

   a. Go to **Core Maintenance → Routing → Service Consumers → RPM_ORIGINATION → Consumer Services tab → OBPY_GET_EXISTING_PARTY**
   Note: If you do not find any Oracle Banking Routing Hub configuration named OBPY_GET_EXISTING_PARTY, that means, the Oracle Banking Routing Hub configurations are not fully imported. Import the Oracle Banking Routing Hub configuration available in the source folder.
   b. From the **Actions**, click on **Request Audit**

   Figure 41: Service Consumers
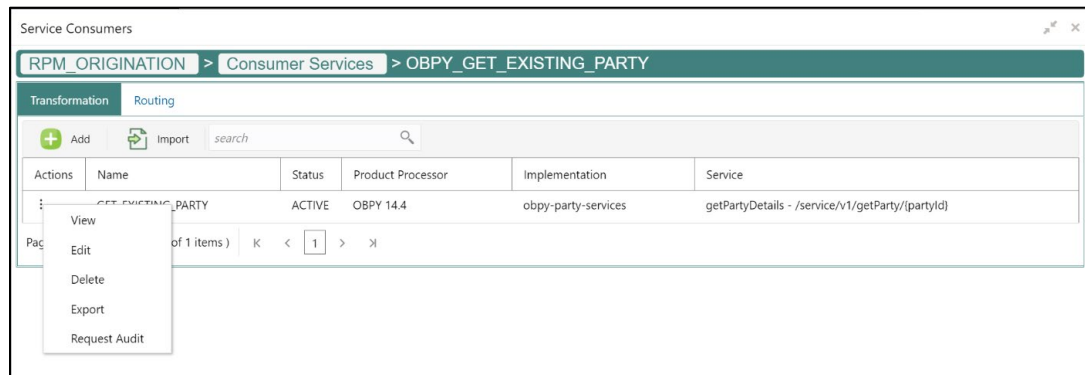
   c. Check the latest **getPartyDetails** Oracle Banking Routing Hub call
   d. Click on the Request Id and check the "Provider Response" to check for any errors

ORACLE®

Figure 42: Request Audit

Figure 43: Request Audit Details

4.3.2 Customer Information Data-segment Dropdowns not Fetching

If in Customer Information data-segment, the existing customer details is not fetching, follow the below steps to

1. Refer **2.2 Preliminary Check for UI** to see if any Party API is failing

2. Check Oracle Banking Routing Hub Audit Request to see if any Oracle Banking Routing Hub calls to Party Module has failed

    a. Go to **Core Maintenance → Routing → Service Consumers → RPM_ORIGINATION → Consumer Services tab → OBPY_MAINTENANCE**
        Note: If you do not find any Oracle Banking Routing Hub configuration named OBPY_MAINTENANCE, that means, the Oracle Banking Routing Hub configurations are not fully imported. Import the Oracle Banking Routing Hub configuration available in the source folder.

    b. From the "**Actions**", click on "**Request Audit**"

    c. Check the latest **getPartyMaintenance** Oracle Banking Routing Hub call

    d. Click on the **Request Id** and click the "**Provider Response**" to check for any errors

4.4 FLEXCUBE Host Integration Troubleshooting

This section describes the possible issues that may occur in FLEXCUBE Universal Banking Solution integration. The possible issues and causes are described in the following subsections:

4.4.1 Host Calls Failing

Host call failure may be due to various reasons ranging from improper Oracle Banking Routing Hub configuration to absence of maintenance in the Oracle FLEXCUBE Universal Banking environment. Host call may fail during Business Product Host Product listing, Interest or Charge Details data-segment fetch or during Oracle FLEXCUBE Universal Banking Account creation time. To find the root issue, follow the below steps

1. Check Oracle Banking Routing Hub Audit Request to see if any Oracle Banking Routing Hub calls to Oracle FLEXCUBE Universal Banking Module has failed

   a. Go to **Core Maintenance → Routing → Service Consumers → RPM_ORIGINATION → FCUBS**
   Note: If you do not find any Oracle Banking Routing Hub configuration for Oracle FLEXCUBE Universal Banking, that means, the Oracle Banking Routing Hub configurations are not fully imported. Import the Oracle Banking Routing Hub configuration available in the source folder.
   b. From the **Actions**, click on **Request Audit**

   Figure 44: Service Consumer
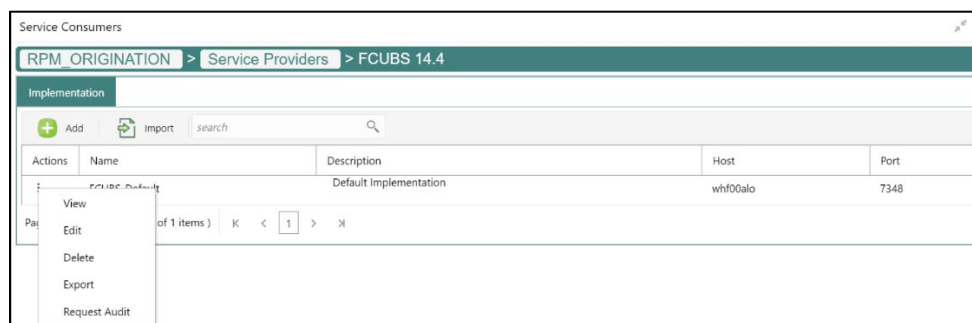
ORACLE®

c. Check the latest Transformation for which you have performed the operation

Figure 45: Request Audit
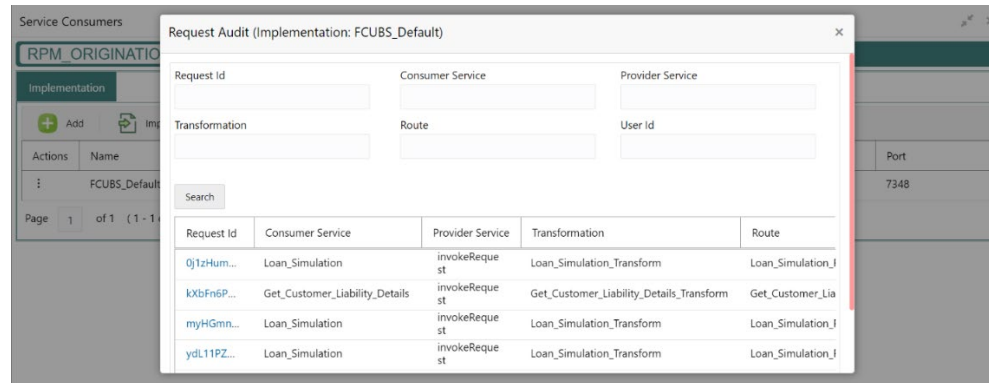
d. Click on the Request Id and check the "Provider Response" to check for any errors

2. If there is no Oracle Banking Routing Hub call itself but still Host call is failing (especially for Account Creation), then failure might be in the workflow task level. In order to debug this scenario, follow the below steps

a. Using the Application Number, call the plato-orch-service search API (API details given below) using Postman
   API Url: http://whf00dtm.in.oracle.com:7011/plato-orch-service/api/v1/extn/custom-actions/queries/tasks?offset=0&limit=100
   Body:

```
{
    "q": "applicationNumber eq 000APP000006967",
        "queryType": [
        "ACQUIRED",
        "AVAILABLE",
        "HOLD",
        "COMPLETED"
    ]
}
```

Headers:
Content-Type:application/json
userId:
appId:platoorch
branchCode:
entityId:DEFAULTENTITY

b. From the response, search for "subWorkflowId"

```
"taskType": "SUB_WORKFLOW",
"status": "COMPLETED",
"inputData": {
    "workflowInput": {
        "TASK_DESCRIPTION": "Savings Origination
            HandOff",
        "applicationDate": 1585218545000,
        "applicationNumber": "000APP000016729",
        "processRefNumber": "000INSTAS0007184",
        "branch": "000",
        "user": "ABIVAN",
        "processName": "INSTANTACCOUNT",
        "processCode": "SavOrig",
        "stage": "Account Creation",
        "stageCode": "RPM_INSTACC_HNDOFF",
        "currentBranchCode": "000"
    },
    "subWorkflowId":
        "ad194dd5-738f-4ce3-b9b9-2a9f72bb59c6",
    "subWorkflowName": "CASAHOSTORCH",
```
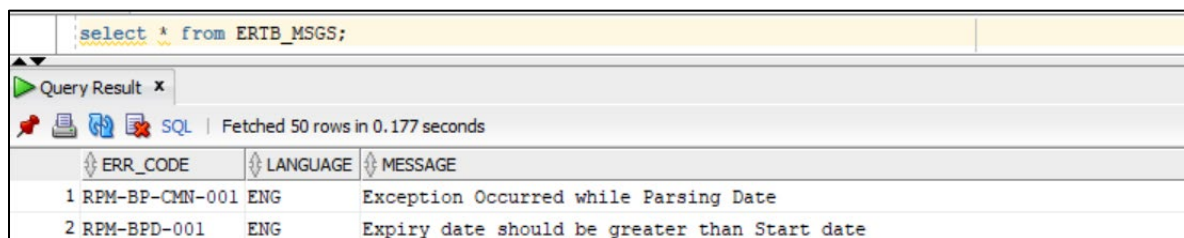
c. Use this subWorkflowId as parameter in the below API
    API Url: http://whf00dtm.in.oracle.com:7011/plato-orch-service/api/workflow/ad194dd5-
    738f-4ce3-b9b9-2a9f72bb59c6
    Headers:
    Content-Type:application/json
    userId:
    appId:platoorch
    branchCode:
    entityId:DEFAULTENTITY
d. The response will show the actual error for HTTP task to fail.

5 Business Error Codes

The list of overrides/information/error codes that might be faced during usage of the application can be found in the table ERTB_MSGS of the corresponding service schema being operated on. For example, if you face an error in Business Product maintenance screen and you want to see the error code in the table, you should connect to your Business Product schema and search for that particular error code in the ERTB_MSGS table.

Figure 46: Error Code and Messages

	ERR_CODE	LANGUAGE	MESSAGE
1	RPM-BP-CMN-001	ENG	Exception Occurred while Parsing Date
2	RPM-BPD-001	ENG	Expiry date should be greater than Start date

```
select * from ERTB_MSGS;
```
Query Result
SQL | Fetched 50 rows in 0.177 seconds

6 Appendix A: Acronyms, Abbreviations and Definitions

The following acronyms/abbreviations are used in this guide:

Table 2: Acronyms and Abbreviations

Acronyms	Definition
SMS	Security Management System
CMC	Common Core
MOC	Mid office Common Core
ELK	Elasticsearch Logstash Kibana

ORACLE®

7 Appendix B: Related Documentation

The related documents are as follows:

- Oracle Banking Common Core User Guide

- Getting Started User Guide

- Oracle Banking Security Management System User Guide

- Alerts and Dashboard User Guide

- Configurations User Guide

- Retail 360 User Guide

- Retail Onboarding User Guide

- Operations User Guide

- Current Account Origination User Guide

- Savings Account Origination User Guide

- Retail Loans Origination User Guide

- Term Deposit Origination User Guide

- Credit Card Origination User Guide

- Tasks User Guide

ORACLE®